**Отчет по практической работе 4**

**Использование встроенных механизмов защиты MS SQL Server**

**Ответить на контрольные вопросы:**

1. Какие существуют встроенные механизмы защиты в MS SQL;
2. В каких случаях их стоит использовать;
3. Как бы вы защищали свои данные.
4. **Какие существую встроенные механизмы защиты MS SQL.**

Самые базовые концепции безопасности SQL Server – **аутентификация** и **авторизация**.

**Аутентификация** — это процесс входа в SQL Server, когда пользователь отправляет свои данные на сервер. Аутентификация устанавливает личность пользователя, который проходит аутентификацию;

**Авторизация** — это процесс определения того, к каким защищаемым объектам может обращаться пользователь, и какие операции разрешены для этих ресурсов.

**Аутентификация в SQL Server**

Аккаунт SQL Server можно разделить на 2 части: **Имя входа** и **Пользователь**.

* **Имя входа** – это глобальный логин для всего экземпляра SQL Server. С помощью него проходится процесс аутентификации;
* **Пользователь** – это участник базы данных, привязанный к определенному Имени Входа.

Например, ваше имя входа на сервер может быть *domain\username*, а пользователь в базе данных, привязанный к этому имени входа может называться *domain\_databaseUser*. Практически всегда имя входа и пользователь в базе данных совпадают по названию, но нужно иметь в виду что они могут и различаться, иметь разные имена.

SQL Server поддерживает 2 режима **аутентификации**:

* **Аутентификация Windows** (Windows Authentication) – аутентификация осуществляется с помощью системы безопасности Windows. Пользователям, которые уже аутентифицированы в Windows и имеют права на SQL Server не нужно предоставлять дополнительные учетные данные.
* **Смешанный режим аутентификации** (Mixed Mode Authentication) – в этом режиме помимо аутентификации Windows поддерживается аутентификация самого SQL Server через логин и пароль.

При Windows аутентификации по сети передаётся серия зашифрованных сообщений, в которых не участвует пароль пользователя. По этому Microsoft рекомендуют использовать именно Windows Аутентификацию.

SQL Server поддерживает три типа **Login Name** (имен входа):

* **Локальная учетная запись** пользователя Windows или учетная запись **домена**/доверенного домена.
* **Группа Windows**. Предоставление доступа локальной группе Windows или группе из AD домена. Позволяет предоставить доступ ко всем пользователям, которые являются членами группы.
* **Логин SQL Server** (SQL Server authentication). SQL Server хранит имя пользователя и хэш пароля в базе данных **master**, используя методы внутренней аутентификации для проверки входа в систему.

**Авторизация в SQL Server**

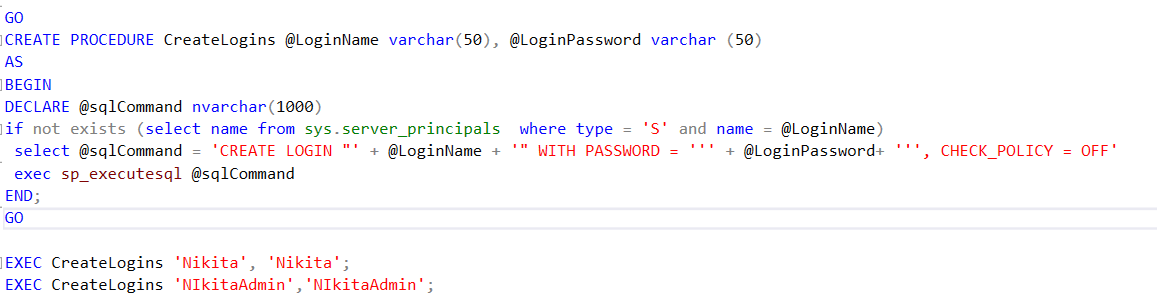
Для авторизации SQL Server использует безопасность на основе ролей, которая позволяет назначать разрешения для роли или группы Windows/домена, а не отдельным пользователям. В SQL Server есть встроенные роли сервера и баз данных, у которых есть предопределенный набор разрешений.

В SQL Server есть 3 уровня безопасности, их можно представить, как иерархию от высшего к низшему:

* **Уровень сервера** – на этом уровне можно раздать права на базы данных, учетные записи, роли сервера и группы доступности;
* **Уровень базы данных** включают в себя схемы, пользователи базы данных, роли базы данных и полнотекстовые каталоги;
* **Уровень схемы** включают такие объекты, как таблицы, представления, функции и хранимые процедуры.

Так как, при сбросе или удалении базы данных сбрасываются или удаляются роли, пользователи этой базы данных. Стоит реализовывать логику создания логинов, схем, пользователей внутри запроса на создание базы данных.

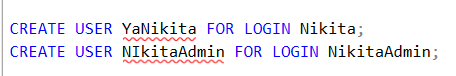
Пример создание учетной записи для авторизации и аутентификации:



Для простоты создания нескольких учетных записей следует создать процедуру, и в дальнейшем, при необходимости добавить учетную запись пользоваться ей.

Пользователи создаются внутри базы данных и привязываются к учетной записи.

Пример:



**Роли приложений**

**Роль** - это определенный набор прав, который можно назначить определенному пользователю или группе пользователей. В SQL Server есть созданные по умолчанию роли уровня сервера и уровня базы данных, которые имеют предопределенный набор разрешений, назначенных для них.

Прмер создания роли и добавления пользователя к роли:





**Шифрование данных средствами SQL Server**

SQL Server может шифровать данные, процедуры и соединения с сервером. Шифрование возможно с использованием сертификата, асимметричного или симметричного ключа. В SQL Server используется иерархичная модель шифрования, то есть каждый слой иерархии шифрует слой под ним. Поддерживаются все известные и популярные алгоритмы шифрования. Для реализации алгоритмов шифрования используется Windows Crypto API.

Самыми распространенными типами шифрования являются TDE (Прозрачное шифрование данных) и Always Encrypted.